

Felipe Canales  
50106507

## **Project 1 Report**

**NOTE:** The results in results.json are out of order -> I couldn't find a way to stop my connected components to stop sorting itself based on label number.... However, the results are still pretty accurate. If you just loop through both json files, you'll see the true accuracy.

### **Enrollment**

For computing features, I was going to use corner detection with Features from Accelerated Segment Test but I couldn't figure out how to compare the keypoints from both images for similarities so I decided to just use SIFT. From the SIFT algorithm, the keypoints and descriptors were calculated. I used those as my features.

### **Detection**

For detecting the different characters in an image, I created a class based off Union Array. This was heavily influenced from [https://www.researchgate.net/publication/235708759\\_Optimizing\\_two-pass\\_connected-component\\_labeling\\_algorithms](https://www.researchgate.net/publication/235708759_Optimizing_two-pass_connected-component_labeling_algorithms) . For my connected components, I used the two-pass algorithm as described in the link above in order to get the labels for each pixel. From there, I computed the max and min values for both the x and y coordinates. This gives us our bounding box. After getting the bounding box, I simply separate the characters in each bounding box into different images and return the image with the bounding box coordinates.

### **Recognition**

For recognition, I used the kth nearest neighbor Brute Force Matcher(BFM) to compare the descriptors of the enrollment image and for every character from the test image. After training the BFM, I used a ratio test to determine which matches were best. The threshold I used for the ratio test was .7. I saved all matches that met the criteria and only selected those that had 2 or more keypoint matches. I then formatted it for the json file.